

Brief a freelance developer

A one-page brief, cost sheet and short guide to ordering a company website or a web application from a freelance developer without paying for parts that are not required.

I have been building web apps since 2019, first as an employee and later as a contractor. The Project Management Institute's 2018 survey of failed projects lists inaccurate requirements gathering and an inadequate vision or goal among the primary causes, and both are settled before the first line of code. Picture a quote based on a vague email: two developers price two different products and the cheaper one is chosen. This is the kind of brief I would like every client to provide me with. Complete it once, send the same document to three freelancers and compare their quotes line by line.

How to use this kit

1. Read the guide below once.
2. Complete the one-page brief. Leave fields blank instead of filling in a guess. A blank field means the developer will ask you a question, and that is fine.
3. Open the cost sheet. Mark the features you need now and the ones that can wait. Keep the sheet next to the quotes you receive.
4. Send the brief to at least three freelancers. Ask each of them to quote the features on your list and not their own interpretation of the project.

Part 1. What a developer needs to know before quoting

The problem, not the solution

Start by describing the problem that exists today. "Our sales team keeps client data in four spreadsheets and nobody knows which one is up to date" is more useful than "we need a CRM". With the first sentence I can propose a shared table with three views. With the second sentence all I can quote is a CRM.

Write down:

- who has the problem (use a role, not a person's name),
- how often it happens,
- what it currently costs you, in hours or lost orders, even as a rough estimate.

Who will use it

List all the types of user for the application and what each of them needs to do. A small business application might have two or three: staff who enter data, a manager who views reports and sometimes clients who log in to view their own orders. Each additional type of user adds development time and testing, so supporting several user types has a significant effect on the price.

The smallest version that is worth paying for

Think about what the application needs to do on day one so that someone stops using the old spreadsheet. That set of features is your first release. Everything else goes on a second list called "later". Developers quote the first list. The second list tells them how to build it so that the "later" features are less likely to need rework.

Data you already have

Say where your data lives today (spreadsheets, an old system, a paper folder) and roughly how much of it there is. The size and condition of the data change the work: moving a small spreadsheet is a small job, while moving ten years of records from an old database with duplicate clients is a separate project. Many customers overestimate the quality of their data and underestimate the effort needed to get it into shape, and data migration is one of the activities that get overlooked or underestimated when a quote is written.

Systems it has to talk to

Payments, invoicing, email, calendars, an accounting tool or a shop. Each connection is a separate task with its own risks: the other system can change its interface or limit how often you can call it. Name each one and say whether it is required or a nice to have.

Accessibility and law

If there are public users, aim for WCAG 2.2 at level AA. The European Accessibility Act applies in the European Union to products and services placed on the market after 28 June 2025, including online shops and banking services, so for those this is not optional. Service providers with fewer than 10 employees and under 2 million euro in annual turnover are exempt. As of October 2026 the standard the EU uses as the reference for the Act, EN 301 549, is still based on WCAG 2.1 AA, and content that conforms to WCAG 2.2 also conforms to 2.1. The W3C reports that accessibility can contribute to cost savings when it is integrated into ongoing development, so ask for it from the start. Also mention personal data. Names and email addresses are personal data under the GDPR, health data is a special category with extra restrictions, and the GDPR requires data protection to be built in when the processing is designed. Say what you store so the developer can plan for it from the start.

Who owns what

Specify in the contract that you own the code, the domain, the hosting account and the database. Also ask for the code to be kept in a repository under your account. Without a signed written transfer, the rights to code written by a contractor can stay with the developer, so this is easy to agree at the start and needs the developer's cooperation to fix later.

Budget and deadline

Give a range, even a wide one. "Between 5,000 and 15,000 USD" lets the developer propose what fits. Without a range the developer has to guess what fits your budget. For the deadline, say what happens if it slips. A trade fair in March is a hard date and "we would like it before summer" is not.

Part 2. How to read the quotes

Ask for a quote per feature

A single number hides what is included. Ask each freelancer to price the features on your list separately, as well as the work that is not a feature: setup, data migration, testing, deployment and handover. When quotes are split this way it is easier to compare them and see where they differ.

Fixed price or time and materials

A fixed price works well when the first release is small and well described. It moves the risk to the developer, who adds a margin for it. Time and materials (an hourly or daily rate) suits anything new or vague, but only with a cap per stage and regular written updates, because the developer has no built-in incentive to keep the hours down. A middle path that several agencies recommend is a short paid discovery stage, followed by a detailed scope and a fixed quote for the build.

What a fair quote includes

- Setup of the project, repository and environments.
- Design of the screens, or a clear note that you provide the designs.
- The features from your list.
- Automated tests for the main business processes, including the parts that handle money, permissions and data.
- Deployment to your hosting account.
- A handover: a short guide on how to run, update and back up the application.
- A warranty on bugs, usually 30 to 90 days.

If any of these items are missing, ask why. Tests and a handover are easy to cut: work near the end of a project is often cut to meet the budget, and leaving out testing or documentation affects the quality of the result and how easily it can be changed later.

Red flags

- No questions before the quote. A developer who quotes your brief without asking anything has probably not read it closely.
- A price far below the others for the same scope. Ask which features were left out.
- Keeping the code on the developer's account "for convenience", with nothing in writing about who owns it.
- No written scope, only a call.
- Full payment up front.

Part 3. Running the project

Milestones and payments

Split the project into a few milestones. Each milestone should end with something you can click through. Tie each payment to the acceptance of a milestone. Freelance web professionals usually ask for a deposit, most often 25 to 50 percent, and larger projects tend to start lower; my own terms are 20 to 30 percent up front. The rest is spread across the milestones, and you can agree to hold back a small final amount until the warranty period ends.

Acceptance

For each feature, write one or two sentences that describe when it is done from the user's point of view: "A manager can export last month's orders to a spreadsheet in under a minute". These sentences are your acceptance tests. They give you and the developer a clear, shared definition of done before the work starts.

Changes during the project

You will probably change your mind, as many teams do: in the Project Management Institute's 2018 survey, 52 percent of projects completed in the previous twelve months had experienced scope creep or uncontrolled changes. Agree on the process for changes before the project starts: a change is written down, priced and either added to the current milestone, moved to "later" or dropped. A change that skips this process is how a project grows past its budget with nobody able to say when it happened.

After launch

Plan a monthly budget for hosting, updates and small fixes. Frameworks and runtimes such as Angular and Node.js publish patch releases often and security fixes several times a year, and each version is supported for a limited time: about 24 months for an Angular major version and 30 months for a Node.js long-term support release. Angular's own guidance is to update one major version at a time, so an app that nobody updates needs more update steps the longer it waits. Ask the developer for a maintenance quote together with the build quote.

Part 4. The one-page brief

Copy this template into a document and fill it in.

Company and contact Company name, website, the person who decides and the person who will test.

The problem today Two to three sentences. Who has it, how often, what it costs.

Users Each type of user and what they need to do.

First release (must have) A numbered list of features, one line each.

Later (nice to have) A numbered list, one line each.

Existing data Where it is, format and approximate size.

Connections to other systems Name, purpose, and whether it is required or nice to have.

Accessibility and personal data Public users or internal only. Personal data stored: yes or no, and what kind.

Design You provide the designs, you have a brand book, or the developer designs.

Ownership Code repository, domain, hosting and database on the company's accounts.

Budget range From and to, with the currency.

Deadline Date, and whether it is hard or soft, with the reason.

How you want to work Fixed price, time and materials with a cap, or paid discovery first. Preferred frequency of updates.

Part 5. The cost sheet

The cost sheet is a spreadsheet with one row for each typical feature. Each row lists what the feature usually involves, the questions that change its size and a column for you to mark "now", "later" or "no". Two columns are left empty for the quotes you receive so that you can compare them side by side.

The sheet does not give prices. Rates vary widely between countries, and a number without your scope would be misleading. What it gives you is the list of things that move the price, so that every quote you compare covers the same ground.

What to send with the brief

- Screenshots of the spreadsheet or system you want to replace.
- Two or three apps or sites you like, with one sentence each on what you like about them.
- Your brand book or logo files, if you have them.

That is enough for a good freelancer to start asking questions and prepare a quote. If you want me to quote your project, send the brief to the address on adrianturbinski.pl and I will reply with questions within two working days.